

## EVALUASI *TEMPLATE MATCHING* PADA PELACAKAN *MARKERLESS* TERHADAP KEMAMPUAN PERANGKAT *SMARTPHONE*

Yudi Setiawan<sup>1</sup>, Kurnia Anggriani<sup>2</sup>, Boko Susilo<sup>3</sup>

<sup>1,2,3</sup>Program Studi Teknik Informatika, Fakultas Teknik, Universitas Bengkulu.  
Jl. WR. Supratman Kandang Limun Bengkulu 38371A INDONESIA  
(telp: 0736-341022; fax: 0736-341022)

<sup>1</sup>ys.teknik@gmail.com

<sup>2</sup>kurniaanggriani@gmail.COM

<sup>3</sup>bksusilo@gmail.com

**Abstrak:** Penelitian ini membahas perancangan aplikasi *Augmented Reality* (AR) untuk melakukan pelacakan *markerless* dengan mengimplementasikan metode *template matching*. Penelitian ini dilakukan bertujuan untuk mengevaluasi metode *template matching* dengan membandingkan tingkat kemampuan perangkat *smartphone* dalam pelacakan *markerless* AR. Evaluasi metode *template matching* berupa identifikasi jarak jangkauan *markerless* dari kamera *smartphone*, sudut jangkauan *markerless* terhadap kamera, dan luas area tangkapan *markerless* yang dapat dikenali oleh aplikasi AR. Ketiga identifikasi *markerless* tersebut kemudian dibandingkan dengan kecepatan *processor*, *read only memory* (RAM), dan besar *pixel* kamera dari berbagai *smartphone* yang berbeda-beda. Aplikasi *Augmented Reality* ini dirancang dengan menggunakan SDK *Vuforia.Markerless* yang dirancang berupa objek-objek *smartphone*. Hasil dari penelitian ini berupa evaluasi dari implementasi metode *template matching* dalam pengenalan *markerless Augmented Reality*, kesesuaian objek sebenarnya dengan informasi virtual yang diberikan (*correctness*), dan pengklasifikasian kemampuan pengenalan *markerless* terhadap kemampuan perangkat *smartphone*.

Kata kunci: *Augmented Reality*, *markerless*, *smartphone*, SDK *Vuforia*.

**Abstract:** This study discusses the design of *Augmented Reality* (AR) application to triangulate *Markerless* by implementing *template matching* method. This research purpose to evaluate the *template matching* method by comparing the level of the tracking capabilities of *smartphone* in *Markerless* AR. *Template matching* method is evaluated by identificate the range of *smartphone* cameras, the angle range of the camera *Markerless* and *Markerless* catchment area that can be recognized by AR applications. Third *Markerless* identification is then compared with the speed of the *processor*, *read only memory* (RAM), and a large *pixel* camera of a variety of different *smartphone*. *Augmented Reality* application is designed by using SDK *Vuforia.Markerless* designed in the form of objects *smartphone*. The results of this research is evaluation of the implementation of *template matching* methods in the introduction of *Augmented Reality Markerless*, conformity with the actual objects given virtual information (*correctness*), and classification capabilities *Markerless* introduction to the capabilities of *smartphone* devices.

**Keywords:** *Augmented Reality*, *Markerless*, *smartphone*, *Vuforia SDK*.

### I. PENDAHULUAN

Salah satu teknik penting dari *digital imagesprocessing* adalah *template matching*. *Template matching* merupakan sebuah teknik yang digunakan dalam mengklasifikasikan sebuah objek dengan melakukan perbandingan bagian gambar dengan gambar lainnya[1]. *Template matching* digunakan secara luas untuk memproses sebuah gambar dan citra. Teknik ini digunakan di dalam aplikasi objek untuk pengalokasian, deteksi tepi objek, untuk memetakan rute untuk *mobile robot* dan pada teknik pendaftaran gambar.

Pada ilmu *Computer Vision*, konsep dari *interest points* disebut juga dengan titik kunci (*keypoints*) atau titik yang memiliki ciri khusus/khas (*feature points*). Sehingga secara umum, *template matching* merupakan metode yang sering digunakan dalam pengolahan citra

pengenalan sebuah objek (*object recognition*). *Template matching* memiliki beberapa metode pelacakan titik (*points detectors*), yaitu; *Harris corners*, *fast feature*, *scale-invariant SURF features*, dan *SURF features*[2].

*Augmented reality* merupakan *synthesis* perumpamaan nyata dan *virtual* [3]. Aplikasi *augmented reality* telah diterapkan di berbagai bidang kehidupan. Perkembangan *smartphone* telah mendukung pengembangan aplikasi ini. *Augmented reality* juga merupakan konsep aplikasi dari menggabungkan dunia fisik (objek sesungguhnya) dengan dunia digital, tanpa mengubah bentuk objek fisik tersebut. Pengenalan objek (teks dan gambar) yang dituju menampilkan berbagai informasi mengenai objek tersebut. *Augmented reality* sebagai sebuah sistem kognitif, hendaknya dapat memahami secara utuh persepsi dari pengguna[4].

Penanda objek pada *Augmented Reality* secara umum dikelompokkan menjadi *marker* dan *markerless*. Penggunaan penanda objek berupa *marker* terdiri dari; *barcode*, *QR Code*, dan *printed AR marker*. Sedangkan penanda objek yang termasuk ke dalam *markerless*, yaitu; *natural printed AR marker* dan *real life marker*. Keuntungan utama dari sistem berbasis *marker* adalah bahwa penanda dapat dirancang sedemikian rupa untuk memastikan mereka tetap relatif baik terdeteksi oleh aplikasi *Augmented Reality*. Kerugian yang jelas dari penggunaan penanda berbasis *marker* adalah modifikasi/perubahan fisik *marker* dari sebuah adegan (*scene*) diperlukan selama proses penangkapan *marker*[5]. Untuk mengurangi ketergantungan terhadap perubahan *marker*, maka penelitian ini menggunakan *markerless* yang memiliki keuntungan perubahan secara natural dari sebuah *sceneAR*.

Penelitian ini, objek *markerless* berupa *smartphone*. Objek *smartphone* digunakan karena *smartphone* memiliki bentuk umum yang sama, yaitu persegi empat atau oval. Akan tetapi, walaupun memiliki bentuk umum yang sama, beberapa *smartphone* memiliki ciri khas/keunikan masing-masing, hal ini yang akan digunakan dalam pengenalan pola *smartphone* menggunakan metode *template matching*.

Paper ini membahas tentang perancangan aplikasi *Augmented Reality* dengan menggunakan *template matching* sebagai metode pelacakan objek *markerless*. Penanda objek yang dirancang berupa *smartphone* sehingga terlebih dahulu akan dilakukan pendefinisian dari pola masing-masing *smartphone*. Kemudian penanda akan diujikan dengan beberapa *smartphone* yang memiliki spesifikasi yang berbeda-beda. Pengujian dilakukan berdasarkan jarak jangkauan pengenalan penanda, besar sudut jangkauan pengenalan penanda, dan luas tangkapan penanda yang dapat dikenali. Dari masing-masing pengujian akan dibandingkan terhadap kemampuan processor, *read only memory*, dan kamera dari masing-masing *smartphone*.

## II. LANDASAN TEORI

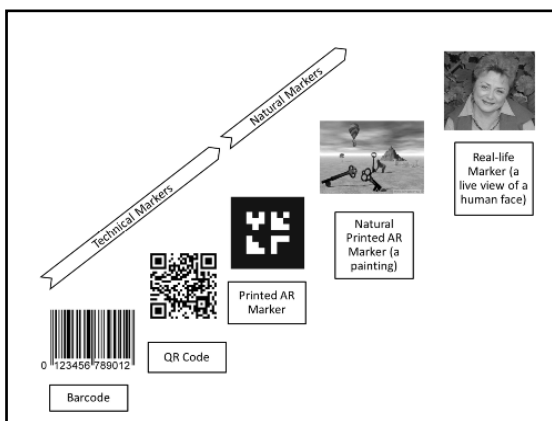
### A. *Augmented Reality*

*Augmented reality* merupakan konsep penggabungan dunia *virtual* ke dalam dunia nyata. Penciptaan dunia *virtual* dilakukan untuk membangkitkan persepsi *user* untuk memahami informasi dari objek yang dikenali. *Augmented Reality* didefinisikan sebagai penggunaan komputer digital *real-time* dan perangkat keras spesial lainnya dan perangkat lunak untuk menghasilkan sebuah simulasi dunia atau lingkungan *alternative*, yang dipercaya sebagai sesuatu yang nyata atau benar bagi *user* [6].

### B. Marker-based dan Markerless AR

Terdapat dua metode pengenalan penanda pada *augmented reality*, yaitu; *marker* dan *markerless*. *Marker* merupakan penanda khusus yang dibuat seperti sebuah *barcode* atau bingkai hitam, sedangkan *markerless* merupakan penanda yang berhubungan dengan objek secara langsung. Adapun evolusi dari penggunaan *marker* hingga penggunaan objek nyata dalam pengenalan penanda *augmented reality* seperti ditunjukkan pada Gambar 1.

Evolusi penanda ditunjukkan dengan perubahan penanda yang dimulai dari bentuk *barcode* hingga bentuk nyata di kehidupan (*real life*). Penanda diklasifikasikan menjadi dua, yaitu; *marker* dan *markerless*. Evolusi penanda yang termasuk di dalam klasifikasi *marker* atau yang dikenal sebagai *technical markers*, yaitu; *barcode*, *QR code*, dan *printed AR marker*. Sedangkan yang termasuk kedalam *markerless* (*natural markers*), yaitu; *natural printed AR marker* dan *real life marker*.



Gambar 1. Evolusi Marker

Penggunaan *marker* atau *markerless* memiliki keuntungan dan kekurangan masing-masing [7]. Adapun keuntungan dari penggunaan *marker*, yaitu:

1. Algoritma pendeteksian yang kecil.
2. Kuat dalam perubahan pencahayaan (stabil).

Adapun kekurangan dari penggunaan *marker*, yaitu;

1. Tidak berkerja pada bagian yang tumpang tindih.
2. Gambar *marker* harus berupa hitam dan putih.
3. Memiliki bentuk persegi yang umum (untuk memudahkan pencarian)
4. *Marker* tidak memiliki keindahan estetika visual.
5. Tidak memiliki objek *real* dalam kehidupan sehari-hari.

Keuntungan dari *markerless*, yaitu:

1. Dapat digunakan untuk mendeteksi objek nyata (*real-world*).
2. Dapat bekerja meskipun target objek di dalam bagian tumpang tindih.
3. Dapat memiliki bentuk dan tekstur yang berubah-ubah (kecuali *solid* dan tekstur gradasi lembut)

### C. Template Matching

Pelacakan titik-titik ciri khas sebuah gambar dengan menggunakan sudut-sudut yang terdapat pada gambar dapat menjadi solusi pencarian pola dari sebuah gambar. Pendeteksian sudut-sudut tersebut dipetakan untuk memudahkan perancangan pola gambar. Sudut-sudut tersebut menjadi menarik karena memiliki dua dimensi yang menjadi ciri yang secara akurat dapat di batasi setiap akurasi sub-*pixel* (sebagai persilangan dua sisi/edges)[2].

*Template matching* merupakan sebuah teknik yang digunakan dalam mengelompokkan sekumpulan objek dengan melakukan perbandingan bagian gambar dengan gambar lainnya. *Template matching* digunakan secara luas untuk memproses sebuah gambar dan citra. Pemilihan pencocokan pola dilakukan secara saling bergantung pada sifat alami dari sebuah gambar

atau objek [1]. Konsep metode pencocokan pola (*template matching*) dapat dilakukan dengan beberapa konsep pelacakan titik, yaitu; *Harris corners*, *fast feature*, *scale-invariant SURF feature*, dan *SURF features*.

Pada penelitian ini, pencocokan pola (*template matching*) menggunakan pelacakan titik *Harris corners*. Hal ini dilakukan dengan menentukan notasi sudut-sudut yang ada pada gambar/objek. Penentu notasi sudut-sudut gambar dilakukan dengan merata-ratakan nilai intensitas perubahan langsung di dalam jendela-jendela kecil sekitar titik perhatian yang diduga. Dengan mempertimbangkan sebuah perpindahan *vector* ( $u,v$ ), maka rerata perubahan intensitas gambar dirumuskan dalam persamaan;

$$R = \sum (I(x+u, y+v) - I(x,y)^2) \quad \dots (1)$$

Persamaan (1) menjelaskan perubahan intensitas titik awal ( $x,y$ ) dengan titik perbandingan ( $u,v$ ) di sekitar titik dugaan awal, yang kemudian dari perubahan intensitas tersebut maka dapat ditentukan titik-titik yang khas sebagai ciri dari gambar tersebut. Sehingga akhir penyelesaian berupa pencarian *pixel-pixel* tetangga sekitar. Perubahan rerata intensitas dapat dikomputasi di dalam semua kemungkinan secara langsung, untuk mendefinisikan sudut sebagai titik yang mana perubahan rerata lebih tinggi dibandingkan di dalam satu arah [2].

### III. PEMBAHASAN

#### A. Perancangan *Markerless*

Perancangan *markerless AR* pada penelitian ini terdiri dari tiga proses perancangan, yaitu; proses pengambilan gambar, proses pelacakan pola gambar, dan proses manajemen database *markerless AR*. *Markerless* dirancang dengan menggunakan gambar dari sisi depan/muka *smartphone*. Proses pengambilan gambar

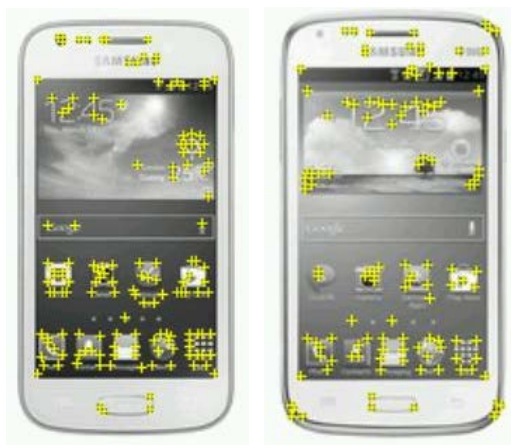
menggunakan kamera digital untuk mendapatkan tekstur dari *smartphone* secara detail. Hasil dari pengambilan gambar sisi depan *smartphone* kemudian dilakukan proses *cropping*, menggunakan *tool* Adobe Photoshop Cs 6, hal ini bertujuan untuk menghilangkan *background*. Jika *background* memiliki tekstur, maka tekstur tersebut akan diperhitungkan sebagai pola nantinya. Adapun gambar sisi depan dari beberapa *smartphone* terdapat pada Gambar 2.



Gambar 2.Sisi Depan *Smartphone*

Gambar sisi depan *smartphone* yang didapatkan, kemudian dilanjutkan dengan proses pelacakan pola dari gambar tersebut. Proses pelacakan pola menggunakan metode *Harris corners* yang akan menghasilkan titik-titik yang merupakan sudut-sudut perubahan intensitas warna dari tekstur gambar. Titik-titik kunci tersebut membentuk pola sebagai pembeda pola *markerless* satu dengan pola *markerless* lainnya. Proses pelacakan pola (*template matching*) dilakukan dengan cara objek-objek pada dunia nyata ditangkap oleh kamera *smartphone*, kemudian dilakukan pengidentifikasian pola dari objek-objek tersebut. Pola dari objek-objek tersebut kemudian akan dilakukan pencocokan dengan pola-pola objek yang telah disimpan di dalam database AR. Adapun gambar pola dari objek yang merupakan

kumpulan dari titik-titik kunci dari beberapa *smartphone* ditunjukkan pada Gambar 3.

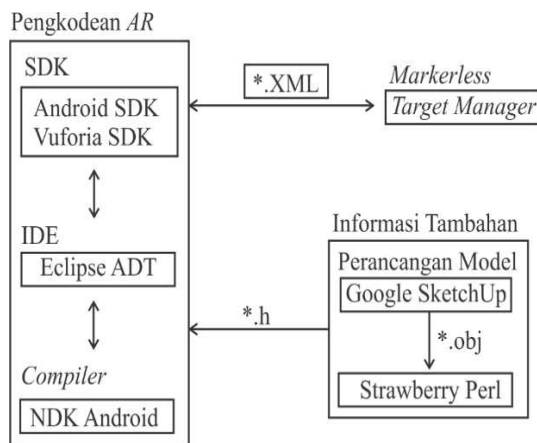


Gambar 3. Pola Objek Smartphone

Penyimpanan informasi dari pola objek *smartphone* dilakukan pada manajemen *database* Android. Manajemen *database* *markerless* AR disimpan di objek *SharedPreferences* dalam format \*.xml. Penyimpanan pola objek berupa penyimpanan koordinat titik-titik kunci dari objek. Koordinat titik-titik kunci tersebut memiliki jarak setiap titik tersebut. Kumpulan titik-titik kunci tersebutlah yang akan membentuk pola sebagai identifikasi objek.

#### B. Perancangan Aplikasi *Augmented Reality*

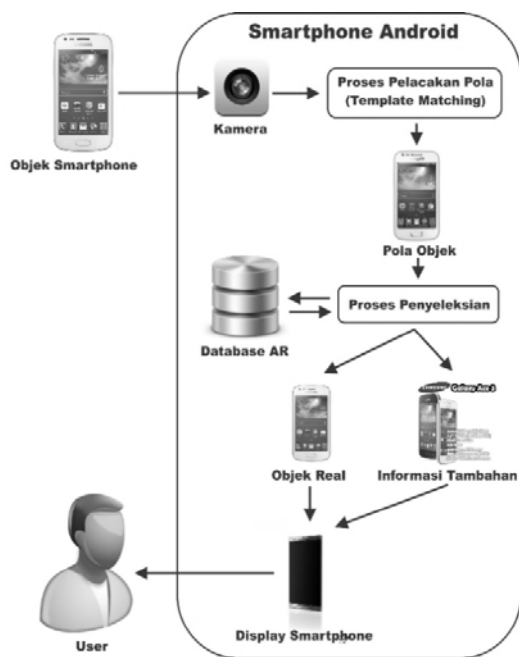
Proses perancangan aplikasi *Augmented Reality*, aplikasi dirancang berbasis Android. Perancangan aplikasi menggunakan SDK Vuforia dengan IDE Eclipse sebagai editor peng-*code*-an. Proses *compiling* dilakukan dengan menggunakan *tool* NDK Android. *Markerless* yang telah dirancang disimpan sebagai objek pengenalan pola dan disimpan dalam format \*.xml. Sedangkan perancangan informasi tambahan dirancang menggunakan aplikasi *Google SketchUp*. Adapun integrasi antar *Augmented Reality Engines* ditunjukkan pada Gambar 4.



Gambar 4. Integrasi Antar *Augmented Reality Engines*

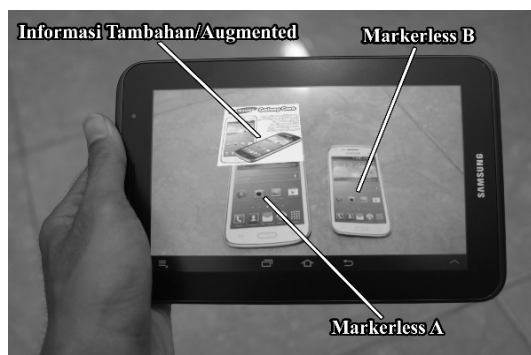
Gambar 4 menunjukkan integrasi antar AR *Engines* yang meliputi proses pengkodean AR, perancangan *markerless*, dan perancangan informasi tambahan dari *markerless*. Pada proses pengkodean AR meliputi penggunaan SDK (Android SDK dan Vuforia SDK), IDE *Eclipse* ADT, dan *Compiler* NDK Android. *Markerless* yang dirancang, terlebih dahulu dilakukan penentuan pola untuk proses pencocokan pola (*template matching*). Pola dari *markerless* kemudian disimpan didalam *database* AR dalam format (di-*export*) \*.xml yang dapat dipanggil oleh Android SDK. Sedangkan untuk informasi tambahan dirancang menggunakan *Google SketchUp* dan dikonversi ke format header \*.h dengan menggunakan *Strawberry Perl*.

Pada Gambar 5 menunjukkan alur kerja dari aplikasi *Augmented Reality* yang dirancang. Proses kerja dari aplikasi dilakukan dengan mengarah kamera *smartphone* ke objek yang sebagai *markerless*. Kemudian objek yang ditangkap dilakukan proses pelacakan pola. Pola objek yang didapatkan dilakukan seleksi dengan pola yang telah disimpan pada *database*, kemudian jika terdapat kecocokan maka akan menampilkan informasi tambahan berdasarkan nama *markerless* tersebut.



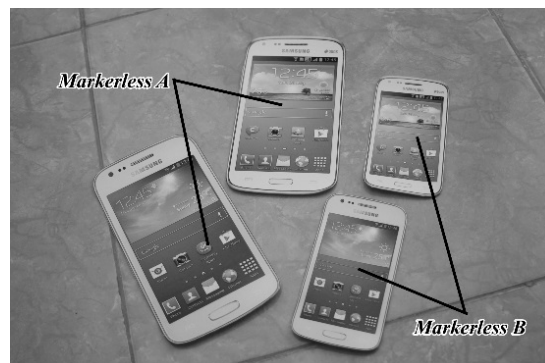
Gambar 5. Alur Kerja Aplikasi AR

Informasi tambahan ditampilkan diatas objek *markerless* yang dapat dilihat oleh *user* pada *display smartphone*. Aplikasi *Augmented Reality* yang dirancang ditunjukkan pada Gambar 6.

Gambar 6. Aplikasi *Augmented Reality*

#### IV. HASIL

Pengujian penelitian ini dilakukan dengan mengukur jarak jangkauan pengenalan *markerless* dari kamera *smartphone*. Jarak jangkauan pengenalan *markerless* diujikan dengan menggunakan 10 *smartphone* dari vendor yang berbeda-beda, dan memiliki spesifikasi yang berbeda-beda pula. Spesifikasi dari 10 *smartphone* tersebut diklasifikasikan berdasarkan besar *pixel* kamera, *clock* Processor, dan RAM. Hasil pengujian jarak pengenalan *markerless* pada *smartphone* terdapat pada Tabel 1. Proses pengujian dilakukan dengan menggunakan dua *markerless* yang berbeda ukuran. *Markerless A* memiliki ukuran dua kali dari ukuran *markerless B*. *Markerless* yang diujikan ditunjukkan pada Gambar 7.

Gambar 7. *Markerless AR*Tabel 1. Hasil Pengujian Jarak Jangkauan Pengenalan *Markerless* Pada *Smartphone*

No	KAMERA (MP)	PROCESSOR	RAM	MARKERLESS A		MARKERLESS B	
				MIN (cm)	MAX (cm)	MIN (cm)	MAX (cm)
1	3.15	Single Core 1GHz	768 MB	10	261	6	188
2	5	Single Core 1GHz	512 MB	9	264	7	193
3	5	Singel Core 1,4GHz	512 MB	9	254	7	178
4	5	Dual Core 1,2GHz	768 MB	9	254	8	201
5	8	Single Core 1 GHz	512 MB	9	259	8	178
6	8	Single Core 1 GHz	512 MB	9	265	6	173
7	8	Single Core 1,4 GHz	512 MB	8	249	7	193
8	8	Dual Core 1,2 GHz	768 MB	9	252	9	183
9	8	Dual Core 1,2 GHz	1 GB	7	234	5	158
10	8	Dual Core 1,4 GHz	1 GB	9	264	6	193

Nilai terendah

Nilai tertinggi

Pengujian jarak dilakukan untuk menentukan jarak jangkauan terdekat untuk kamera mengenali *markerless* (jarak minimum), dan jarak jangkauan terjauh untuk kamera mengenali *markerless* (jarak *maximum*). Jarak jangkauan terendah (*minimum*) yang baik adalah jarak jangkauan yang memiliki nilai terendah, karena *user* dapat mengetahui informasi tambahan objek tersebut pada jarak yang dekat. Sedangkan untuk jarak jangkauan terjauh (*maximum*) yang baik adalah jarak jangkauan yang memiliki nilai tertinggi, karena dapat membantu *user* untuk mendapatkan informasi tambahan objek tersebut pada jarak yang jauh.

Pengukuran jarak jangkauan pengenalan *markerless* dilakukan dengan meletakkan *markerless* tepat di depan perangkat Android, yang telah terpasang aplikasi *Augmented Reality* yang telah dirancang dengan besar sudut  $90^{\circ}$ . Adapun posisi pengujian *markerless* terhadap perangkat Android ditunjukkan pada Gambar 8.



Gambar 8. Proses Pengujian Jarak Jangkauan Pengenalan *Markerless*

Proses pengujian dilakukan dengan mengukur jarak jangkauan dimana aplikasi AR masih dapat melakukan pelacakan dan pengenalan *markerless*. *Markerless* diletakkan tegal lurus di depan perangkat Android. Adapun alat ukur yang digunakan berupa penggaris yang diletakkan di samping perangkat Android.

Nilai jarak jangkauan hasil pengujian yang didapatkan, kemudian dimasukkan ke dalam tabel untuk menentukan nilai ralat mutlak, ralat nisbi, keseksamaan, dan data pengukuran. Pencarian nilai ralat mutlak dimaksudkan untuk mengukur ketidakpastian akibat kesalahan pengukuran ataupun kesalahan kalibrasi.

Proses pelacakan titik-titik *features* terhadap resolusi kamera didapatkan dari hasil pembentukan matriks dan proses pemetaan titik-titik *features* pada matriks. Pembentukan kolom dan baris pembentuk matriks dilakukan pada fungsi *void configureVideoBackground()*. Kolom dan baris matriks didapatkan dari:

1. Resolusi kamera *smartphone* (lebar dan tinggi).
2. Resolusi layar *smartphone* (lebar dan tinggi).

Kemudian dari resolusi kamera dan layar *smartphone* dilakukan perhitungan pada formula (2), (3), (4), dan (5) sebagai berikut:

Kolom matriks:

$$(\text{Data}[0]) = \frac{\text{tinggi kamera} \times (\text{tinggi layar} / \text{lebar kamera})}{\dots} \quad (2)$$

Baris matriks:

$$(\text{Data}[1]) = \text{tinggi layar} \quad \dots (3)$$

Jika kolom matriks ( $\text{Data}[0]$ ) < baris matriks ( $\text{Data}[1]$ ), maka;

Kolom matriks baru:

$$(\text{Data}'[0]) = \text{lebar layar} \quad \dots (4)$$

Baris matriks baru:

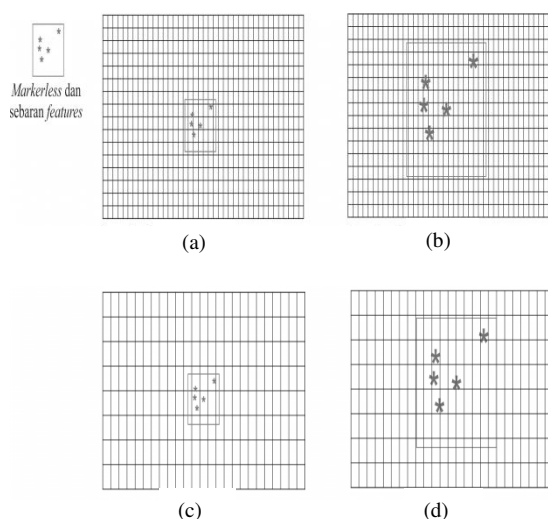
$$(\text{Data}'[1]) = \frac{\text{lebar layar} \times (\text{lebar kamera} / \text{tinggi kamera})}{\dots} \quad (5)$$

Adapun kolom dan baris matriks, dan jumlah *cell* matriks dari *smartphone* yang diujikan pada penelitian ini terdapat pada Tabel 2.

Tabel 2. Baris Dan Kolom Pembentuk Matrik Dari *Smartphone*

No	AF/ Non	Cam	Video		Screen		DATA [0]	DATA [1]	DATA' [0]	DATA' [1]	Cell
			Height	Width	Width	Height					
1	Non	3.15	2048	1536	320	480	640	480	640	480	307,200
2	AF	5	2592	1944	480	800	1066.67	800	1066.67	800	853,333
3	AF	5	2592	1944	480	800	1066.67	800	1066.67	800	853,333
4	AF	5	2592	1944	480	800	1066.67	800	1066.67	800	853,333
5	Non	8	2596	1948	480	800	1066.12	800	1066.12	800	852,895
6	AF	8	3264	2448	480	854	1138.67	854	1138.67	854	972,421
7	AF	8	3264	2448	480	854	1138.67	854	1138.67	854	972,421
8	AF	8	3200	2400	540	960	1280	960	1280	960	1,228,800
9	AF	8	3264	2448	540	960	1280	960	1280	960	1,228,800
10	AF	8	3264	2448	800	1280	1706.67	1280	1706.67	1280	2,184,533

Pemetaan matriks yang dibentuk dan pemetaan *featuresmarkerless* ditunjukkan pada Gambar 9.

Gambar 8. Pemetaan Titik *Features* Pada Matriks Yang Dibentuk

Pada Gambar 8 ditunjukkan bahwa:

1. Gambar 8.a, penanda pada posisi jauh, terhadap matriks dengan kolom dan baris yang besar (rapat), maka titik-titik *features* dapat dipetakan satu *feature* pada satu *cell*.
2. Gambar 8.b, penanda pada posisi dekat terhadap matriks dengan kolom dan baris yang besar (rapat), maka titik-titik *features* dapat dipetakan satu *feature* pada satu *cell*.
3. Gambar 8.c, penanda pada posisi jauh terhadap

matriks dengan kolom dan baris yang sedikit (renggang), maka titik-titik *features* memungkinkan terdapat lebih dari satu *feature* pada satu *cell*.

4. Gambar 8.d, penanda pada posisi dekat terhadap matriks dengan kolom dan baris sedikit (renggang), maka titik-titik *features* dapat dipetakan satu *feature* pada satu *cell*.

Analisis hasil pengujian kebutuhan (*requirements testing*) sistem terhadap resolusi kamera *smartphone*; *smartphone* dengan resolusi kamera rendah membentuk pemetaan gambar dengan ukuran kolom dan lebar matriks yang lebih kecil dibandingkan dengan pemetaan gambar oleh *smartphone* dengan resolusi kamera tinggi. Hal ini menyebabkan pada *smartphone* dengan resolusi kamera rendah dapat memetakan gambar untuk melacak seluruh *features* dari *markerless* pada jarak minimum, dan melakukan pelacakan *features markerless* tepat pada masing-masing kolom dan baris matriks. Sedangkan, pemetaan gambar dalam matriks berukuran besar memudahkan untuk pelacakan *markerless* pada jarak jangkauan *maksimum*, karena pelacakan *features* lebih rapat dan dapat dipetakan pada setiap *cell matriks*.



Tabel 3 menunjukkan jarak jangkauan pengenalan *markerless* berdasarkan besar *pixel* kamera. Hasil pengujian menunjukkan bahwa kamera dengan *pixel* yang rendah (3,15MP → 261cm dan 5MP → 190,7cm) dapat mengenali *markerless* untuk jarak yang jauh dibandingkan kamera dengan *pixel* tinggi (8MP). Sedangkan, kamera dengan *pixel* yang tinggi (8MP → 8,5cm dan 6,8cm) dapat mengenali *markerless* untuk jarak yang dekat dibandingkan kamera dengan *pixel* rendah (3,15MP dan 5MP).

Tabel 3. Pengujian Jarak Jangkauan Pengenalan *Markerless* Berdasarkan Besar *Pixel* Kamera

KAMERA	MARKERLESS A		MARKERLESS B	
	MIN	MAX	MIN	MAX
3.15 MP	10	261	6	188
5 MP	9	257,3	7,3	190,7
8 MP	8,5	253,8	6,8	179,7

Pada Tabel 4, pengujian jarak jangkauan pengenalan *markerless* berdasarkan nilai *clock* Processor, menunjukkan bahwa *clock* Processor yang tinggi (*dual core* 1,2GHz → 8,3cm, dan *dual core* 1,4GHz → 6cm) dapat mengenali *markerless* untuk jarak yang dekat dibandingkan dengan *clock* Processor rendah (*single core* 1GHz dan *single core* 1,2GHz). Sedangkan untuk jarak jangkauan pengenalan yang jauh, Processor dengan *clock* yang tinggi (*dual core* 1,4GHz → 264cm dan 193cm) lebih jauh dibandingkan Processor dengan *clock* rendah (*single core* 1GHz, *single core* 1,2GHz, *dualcore* 1,2GHz).

Tabel 4. Pengujian Jarak Jangkauan Pengenalan *Markerless* Berdasarkan *Processor*

PROCESSOR (GHz)	MARKERLESS A		MARKERLESS B	
	MIN	MAX	MIN	MAX
Single Core 1	9,25	262,3	6,8	183
Single Core 1,2	8,5	251,5	7	185,5
Dual Core 1,2	8,3	246,7	7,3	180,7
Dual Core 1,4	9	264	6	193

Pada Tabel 5, pengujian jarak jangkauan pengenalan *markerless* berdasarkan besar RAM, menunjukkan bahwa RAM yang tinggi (1GB → 8cm dan 5,5cm) dapat mengenali *markerless* pada jarak yang dekat dibandingkan RAM yang rendah (512MB dan 768MB), sedangkan RAM yang rendah (512MB → 258,2cm, 768MB → 190,7cm) dapat mengenali *markerless* untuk jarak yang jauh dibandingkan dengan RAM yang tinggi (1GB)

Tabel 5. Pengujian Jarak Jangkauan Pengenalan *Markerless* Berdasarkan RAM

RAM	MARKERLESS A		MARKERLESS B	
	MIN	MAX	MIN	MAX
512 MB	8,8	258,2	7	183
768 MB	9,3	255,7	7,7	190,7
1 GB	8	249	5,5	175,5

## V. KESIMPULAN

Penelitian ini melakukan evaluasi metode *template matching* pada pengenalan *markerless Augmented Reality*. Evaluasi tersebut meliputi jarak jangkauan pengenalan *markerless* terhadap besar *pixel* kamera, *clock* Processor, RAM, dari beberapa *smartphone* yang berbeda-beda. Jarak jangkauan pengenalan *markerless* terdekat (*minimum*) dapat dilakukan dengan baik pada *smartphone* dengan besar *pixel* kamera tinggi, Processor *clock* tinggi dan RAM yang tinggi. Sedangkan, untuk mengenali penanda terjauh (*maximum*) dapat dilakukan dengan baik pada besar *pixel* kamera rendah, *clock* Processor tinggi, dan RAM yang rendah.

## REFERENSI

- [1] M.T., M.R., and S.P, "Review Article: An Overview of Template Matching Technique In Image Processing," *Reaserch Journal of Applied Sciences Engineering and TEchnology* 4(24):5469-5473 2012, 2012.
- [2] R. Laganier, *OpenCV 2 Computer Vision Application Programming Cookbook* vol. 1. BIRMINGHAM-MUMBAI: PACKT Publishing Ltd., 2011.

- [3] G. Klein, "Visual Tracking For Augmented Reality," Degree of Doctor of Philosophy, Department of Engineering, University of Cambridge, 2006.
- [4] D. Stricker and G. Bleser, "From Interactive to Adaptive Augmented Reality," in *Ubiquitous Virtual Reality (ISUVR), 2012 International Symposium on*, 2012, pp. 18-21.
- [5] C. R. Warrington, "Markerless Augmented Reality for Panoramic Sequences," Postgraduate Program, School of Information Technology and Engineering, Faculty of Engineering, University of Ottawa, Canada, 2007.
- [6] M. Hincapie., A. Caponio., H. Rios., and E. G. Mendivil, "An Introduction to Augmented Reality with Applications in Aeronautical Maintenance," in *ICTON 2011*, 2012.
- [7] D. L. Baggio, D. M. Escrivá, N. Mahmood, R. Shilkrot, S. Emami, K. Levgen, and J. Saragih, *Mastering OpenCV with Practical Computer Vision Projects*: PACKT Publishing, 2011.